FPT UNIVERSITY

FPT UNIVERSITY HCMC

Face recognition and tracking in video

Computer Vision - SU24

Lê Anh Tuấn— SE173591

Nguyễn Bảo Hân— SE182364

Trần Hoàng Nam— SE182380

1. Introduction

In recent years, the rapid advancement of the computer vision specifically in research and development. One of the fundamental task is face recognition and tracking in video streams. Biometric systems, including face, iris, fingerprint, retina, palm, play an important role in authentication and identification because of unique personal characteristics [1].
Face recognition technology has emerged as a pivotal solution in various real-life applications. For instance, some real-life applications that can be mentioned include: Check-in System Based on Face Recognition Technology [3], Automatic Classroom Attendance Management System [6],... Face recognition system is automatically identify a person from an image, video, ... use complex algorithms to examine and contrast facial characteristics. It recognizes facial features by taking features out of a picture of the subject's face and examining things like size, shape, and relative position. Concurrently, face tracking focuses on the continuous localization and monitoring of faces over a series of video frames enabling real-time spatial awareness and interaction.
In this report, we apply a machine learning technique called Support Vector Classification (SVC) to improve the precision and resilience of face recognition and tracking systems. SVC's ability to identify complex data patterns makes it useful for face feature extraction and classification. Our goal is to enhance the precision and resilience of our face recognition and tracking approach by including SVC.

2. Problem Definition

Face recognition and tracking in video are essential tasks in computer vision, with applications ranging from surveillance to human-computer interaction. This section outlines the specific scope of our project and defines the problem we aim to address.
Our project focuses on real-time face recognition and tracking within video streams. We aim to develop an efficient system capable of:

- Automatically identifying faces in video frames;
- Associating detected faces with known identities;
- Continuously tracking the same face across consecutive frames, despite challenges such as occlusions, scale changes, and abrupt movements.

2.1. Face Detection and Recognition

The system should automatically detect faces in video frames with high accuracy. Once faces are detected, the system must reliably associate them with known identities from a database. Key challenges include variability in lighting conditions, diverse facial poses, and expressions. The system should be robust enough to handle partial face visibility and occlusions, and it must efficiently manage large-scale face databases to ensure real-time performance.

2.2. Face Tracking

The system should continuously track faces across consecutive frames, maintaining identity consistency. It needs to overcome challenges such as motion blur, scale variations, and sudden movements, and effectively handle occlusions and temporary disappearances of faces. Ensuring real-time performance is critical for practical applications.

3. Method

The method employed for this study is Support Vector Classification (SVC), a supervised machine learning algorithm that is widely used for classification tasks.
Classification is a methodical grouping of data into meaningful categories based on similarities, using codes and descriptors to organize survey responses effectively. This approach is crucial for developing statistical surveys. A classifier, an abstract metaclass, categorizes instances sharing common features.
Support Vector Machine (SVM) stands out as one of the most effective machine learning algorithms, pio neered by Vapnik in the 1990s. SVMs encompass a family of supervised learning methods for both classification and regression tasks. Supervised learning involves deducing patterns from labeled training data, where algorithms derive an inferred function known as a classifier.
SVMs are prized for their high accuracy and robust theoretical underpinnings against overfitting. They excel with suitable kernels, allowing effective handling of non-linearly separable data within the feature space. This capability makes SVMs a preferred choice, particularly in scenarios like face recognition and video tracking, where precise classification of complex visual data is paramount.
SVC works by finding the optimal hyperplane that separate data into different classes created to optimize the margin in the training data between several classes. The distance between the hyperplane and the support vectors—the nearest data points from each class—defines the margin. The hyperplane that maximizes this margin is the ideal one. The distance d from a point $\mathbf{x}_0 = (x_{0,1}, x_{0,2}, \ldots, x_{0,n})$ to the hyperplane is given by:

$$d = \frac{|\mathbf{w} \cdot \mathbf{x}_0 + b|}{\|\mathbf{w}\|}$$

where:

- $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ is the weight vector (normal vector) of the hyperplane,
- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is a point on the hyperplane,
- b is the bias term,

3.1. SVM Classification

For this study, we classify facial into categories using a Support Vector Machine (SVM) classifier. SVMs are particularly suitable for this task due to their ability to handle high-dimensional feature spaces efficiently and their robustness against the curse of dimensionality.

The decision function $f(\mathbf{x})$ learned by SVM is described as:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

where $\mathbf{w}$ is the weight vector, b is the bias term, and $\mathbf{x}$ is the input sample.

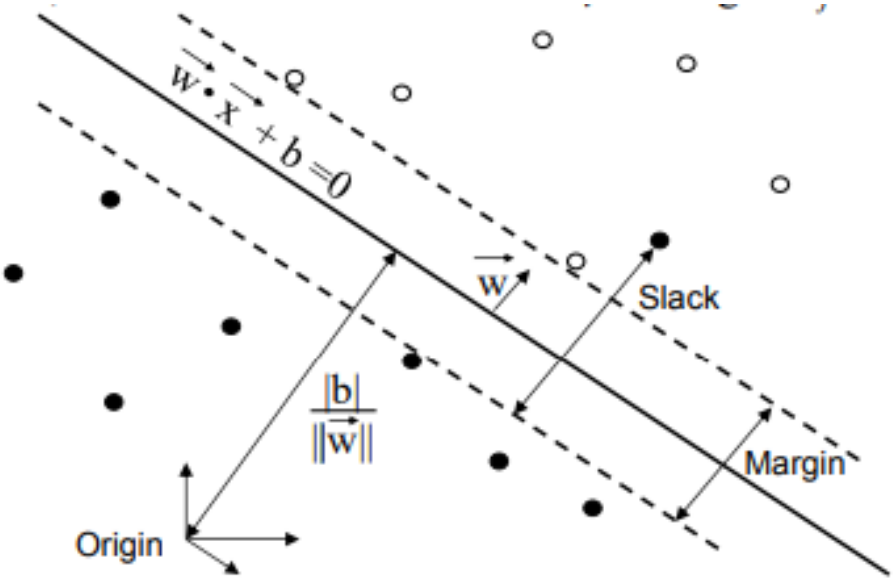

Figura 1: SVM Classifier, maximization of margin.

3.2. Data Collection and Preprocessing

We collected a dataset comprising facial images from various individuals sourced from publicly available repositories. Each image was annotated with the respective person's name to facilitate later classification and model training. The images were captured under controlled lighting conditions to ensure consistent image quality.

3.3. Landmark Extraction using Mediapipe Face Mesh

We utilized Mediapipe Face Mesh to extract facial landmarks from the images. Mediapipe Face Mesh provides 468 feature points per face, including x, y, z coordinates for each point. This process enabled us to represent facial images as numerical feature vectors, suitable for subsequent machine learning model training.

3.4. Feature Engineering and Model Training

Following landmark extraction with Mediapipe Face Mesh, we prepared the data for SVM model training. Initially, we employed label encoding to convert the names of individuals into numeric indices. Subsequently, we split the data into training and testing sets with an 80-20 ratio. To enhance accuracy and consistency, we applied feature scaling to normalize the feature values.

3.5. Implementation Details

The entire research and implementation were conducted using the Python programming language, leveraging libraries such as OpenCV, Mediapipe, and sklearn. These tools facilitated functions ranging from image processing and feature extraction to model training and evaluation within the research framework.

3.6. Kernel Selection

SVM operates by mapping input data into a high-dimensional feature space where it finds the optimal hyperplane that separates different classes of data points. We chose a linear kernel $K(x, x') = x^T x'$ for our SVM model due to its simplicity and effectiveness in handling our high-dimensional feature vectors. The linear kernel computes the dot product between input features, making it suitable for our numerical feature vectors derived from facial landmarks.

3.7. Model Training

To train the SVM model, we used a dataset consisting of labeled facial images. Each image was associated with a numeric label representing the individual's identity. We split the dataset into training and testing sets, using 80% of the data for training and 20% for testing. During training, SVM learns the optimal hyperplane that maximizes the margin between different classes of facial feature vectors.
The objective function of the SVM can be formulated as:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$

subject to:

$$y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

where w is the weight vector, b is the bias term, ξ_i are slack variables, $\phi(x_i)$ is the feature mapping of input x_i, and C is the regularization parameter controlling the trade-off between maximizing the margin and minimizing classification errors.

3.8. Performance Evaluation

After training, we evaluated the SVM model using performance metrics including accuracy, classification report, and R^2 score. Accuracy measures the percentage of correctly classified instances, while the classification report provides detailed metrics such as precision, recall, and F1-score for each class. The R^2 score indicates the proportion of variance in the dependent variable (label) that is predictable from the independent variables (features).
In our application, we are using the RBF Gaussian kernel, which is suitable with the current number of features. Also, it gives the best results compared to other kernels. For the optimization, the kernel width and the penalty parameter C are set to $\sigma = 3$ and $C = 5$, which have been determined through cross-validation. For more details, the reader may refer to [4]. Generally, SVM is sensitive to the scaling problem. This challenge is tackled by the use of normalized feature data for training and testing.

According to (1), the binary SVM classifier returns whether the input sample belongs to a particular class or not. As we are dealing with a multi-class problem, we further compute the probability p_j for every class j based on the training model by applying the method of pairwise coupling [5]. Here, the libSVM implementation has been used for software realization [2].

4. Implementation and Results

In this section, we present the experimental evaluation of a real-time face recognition system utilizing webcam-captured images. The system leverages machine learning techniques and real-time image processing to predict labels corresponding to recognized faces.
The evaluation involved real-time prediction of labels for facial images captured directly from a webcam. The system was trained using Support Vector Machine (SVM) algorithms and MediaPipe's face mesh detection for facial landmark extraction. This setup enabled comprehensive analysis across various facial expressions and environmental conditions.
Each webcam frame was processed using MediaPipe's Face Mesh to extract facial landmarks. These landmarks were then fed into a pre-trained SVM model to predict the identity of the detected face. The prediction process occurred in real-time, allowing for instantaneous recognition and labeling of faces within the camera's view.
The experimental results demonstrated robust performance of the face recognition system. Real-time predictions were consistently accurate across different facial poses, lighting conditions, and backgrounds. The SVM model, trained on a dataset encompassing diverse facial features, exhibited high accuracy in identifying individuals in dynamic settings.
The outcomes highlight the efficacy of combining SVM-based classification with MediaPipe's face mesh for real-time face recognition tasks. The system's ability to swiftly and accurately predict labels from webcam-captured images underscores its practical applicability in interactive and security-sensitive environments.

5. Reference

References

[1] Muhammet Baykara and Resul Daş. Real time face recognition and tracking system. pages 159–163, 2013. pages 2

[2] C. C. Chang and C. J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):Article 27, 2011. pages 6

[3] Y. Chen, H. Zhang, D. Wu, and J. Gong. Check-in system based on face recognition technology. *Journal of Physics: Conference Series*, 1656(1):012046, 2020. pages 2

[4] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. pages 5

[5] T. Wu, C. Lin, and R. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004. pages 6

[6] X. Yan and J. Ye. Automatic classroom attendance management system. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2965–2973, 2017. pages 2